KLASIFIKASI DATA MENGGUNAKAN JST BACKPROPAGATION MOMENTUM DENGAN ADAPTIVE LEARNING RATE

Warih Maharani
Fakultas Teknik Informatika, Institut Teknologi Telkom
Jl. Telekomunikasi No.1 Bandung 40286
Telp. (022) 7564108 ext 2104, Faks. (022) 7565931
E-mail : wrh@ittelkom.ac.id

Abstrak
Data mining merupakan suatu proses pengekstrakan informasi penting pada data yang berukuran besar. Salah satu fungsionalitas yang sering digunakan pada data mining adalah klasifikasi, yang berfungsi menemukan sekumpulan model/fungsi sehingga dapat membedakan kelas data untuk keperluan prediksi. Jaringan Syaraf Tiruan (JST), merupakan salah satu teknik klasifikasi yang cukup handal dikarenakan kemampuannya dalam memprediksi. JST mempunyai toleransi yang tinggi terhadap data yang mengandung noise serta bersifat adaptive, dimana jaringannya mampu belajar dari data yang dilatihkan kepadanya. Oleh karena itu penelitian ini menganalisis pengklasifikasian data dengan menggunakan JST Backpropagation Momentum dengan adaptive learning rate untuk mendapatkan hasil yang optimal.
Sebelum memasuki tahap klasifikasi, proses yang dilakukan adalah feature selection. Feature selection merupakan tahap preprocessing yang bertujuan untuk mencari atribut yang relevan terhadap label kelas. Dengan kata lain, feature selection dapat dikatakan sebagai teknik mereduksi dimensi sebagai usaha untuk meningkatkan performansi dari sebuah classifier. Metode feature selection yang digunakan adalah information gain. Setelah dilakukan preprocessing data, kemudian dilakukan tahap klasifikasi menggunakan JST Backpropagation Momentum dengan adaptive learning rate..
Hasil pengujian menunjukkan bahwa dengan adanya konstanta momentum dan adaptive learning rate mempercepat kecepatan belajar jaringan. Selain itu juga berpengaruh terhadap nilai keakuratan sehingga dapat mencapai tingkat akurasi sebesar 96%.

Keyword : *Jaringan Syaraf Tiruan, backpropagation momentum, adaptive learning rate, feature selection*

1. PENDAHULUAN

1.1 Latar Belakang

Salah satu manfaat dari data mining adalah untuk mengklasifikasikan data. Jaringan Syaraf Tiruan (JST) merupakan salah satu metode klasifikasi yang meniru cara kerja otak manusia yang dapat menyelesaikan suatu permasalahan dengan cara pembelajaran (*learning*). Kelebihan JST salah satunya adalah kemampuannya dalam beradaptasi sehingga mampu belajar dari data masukan yang diberikan sehingga dapat memetakan hubungan antara masukan dan keluarannya. Selain itu kemampuan JST dalam memprediksi keluaran berdasarkan masukan yang telah dilatihkan sebelumnya [4].
Akan tetapi, tidak semua data memiliki pengaruh yang sama untuk mengklasifikasikan data. Sehingga diperlukan sebuah proses untuk menentukan nilai *informative* dari suatu atribut, salah satunya adalah metode *feature selection*. Nilai *informative* tersebut merepresentasikan seberapa besar kontribusi suatu atribut pada proses klasifikasi data. Penelitian ini menganalisis penggabungan metode *feature selection* dan JST dimana untuk algoritma pembelajarannya dipilih algoritma *Backpropagation Momentum* dengan *adaptive learning rate* untuk mendapatkan hasil yang optimal.

1.2 Permasalahan

Berdasarkan latar belakang tersebut, permasalahan yang muncul adalah :

a. bagaimana melakukan *discretization* pada data
b. bagaimana menentukan *information gain* pada metode *feature selection*
c. bagaimana pemodelan *classifier* menggunakan JST *Backpropagation Momentum* dengan *adaptive learning rate*
d. bagaimana menganalisis klasifikasi berdasarkan parameter-parameter masukannya

1.3 Tujuan

Tujuan penelitian ini adalah melakukan klasifikasi data dengan menggunakan JST *Backpropagation Momentum* dengan *adaptive learning rate* yang digabungkan dengan metode *feature selection*. Kemudian dilakukan analisis berdasarkan akurasi sistem klasifikasi yang dibangun.

2. TINJAUAN PUSTAKA

2.1 Data Mining

Data mining adalah suatu proses pengekstrakan informasi penting pada data yang berukuran besar. Beberapa fungsionalitas pada data mining, diantaranya adalah [1] :

a. Deskripsi kelas
   Meringkas sekumpulan data yang didapat dari karakterisasi data atau diskriminasi data.
b. Analisis asosiasi
   Mendeskripsikan pola-pola yang muncul secara bersamaan dalam suatu data menggunakan aturan asosiasi
c. Klasifikasi dan prediksi
   Membuat suatu model atau fungsi sehingga dapat menggambarkan dan membedakan kelas data dengan tujuan agar model tersebut dapat digunakan untuk memprediksi kelas dari objek data yang belum diketahui kelasnya
d. Analisis *cluster*
   Mengelompokkan sekumpulan objek ke dalam kelas yang sama dengan memaksimalkan kemiripan satu sama lain pada objek-objek dengan kelas sama dan tidak mirip dengan objek pada cluster lain.
e. Analisis *outlier*
   Outlier merupakan data yang tidak umum dan tidak sesuai dengan model data pada umumnya

2.2 Klasifikasi

Klasifikasi adalah proses menemukan sekumpulan model/fungsi yang menjelaskan dan membedakan data kedalam kelas-kelas tertentu, dengan tujuan menggunakan model tersebut dalam menentukan kelas dari suatu objek yang belum diketahui kelasnya[1].
Pada 2 proses dalam klasifikasi, yaitu [1] :

a. Proses *learning/training*
   Melakukan pembangunan model menggunakan data *training*. Pada penelitian ini menggunakan model JST.
b. Proses *testing*
   Melakukan tes terhadap data *testing* menggunakan model yang telah diperoleh dari proses *training*.

2.3 Feature Selection

Feature selection merupakan tahap *preprocessing* yang bertujuan untuk mencari nilai relevansi suatu atribut terhadap label kelas dan mengabaikan atribut yang tidak memberikan kontribusi apapun terhadap klasifikasi data. Ini merupakan teknik mereduksi dimensi untuk meningkatkan performansi dari sebuah *classifier*.
Pada JST, *feature selection* berpengaruh terhadap pembentukan arsitektur yang dibuat. Jika *feature*/atribut yang digunakan dikurangi dimensinya, maka jumlah input neuron yang digunakan juga berkurang. Hal ini terjadi karena jumlah atribut pada JST identik dengan jumlah input neuronnya.

2.4 Information Gain

Penerapan *feature selection* pada penelitian ini adalah dengan menghitung *information gain* pada setiap atributnya. *Information gain* dari suatu atribut, diperoleh dari nilai *entropy* sebelum pemisahan dikurangi *entropy* setelah pemisahan.
Berikut adalah proses perhitungan *information gain* [4] :

a. Cari nilai entropy sebelum pemisahan dengan cara berikut :
   $$Entropy(S)=\sum_{i=1}^{k}(P_i)\log 2(P_i) \quad (1)$$
   Dengan Pi adalah proporsi data S dengan kelas i, dan k adalah jumlah kelas pada output S.

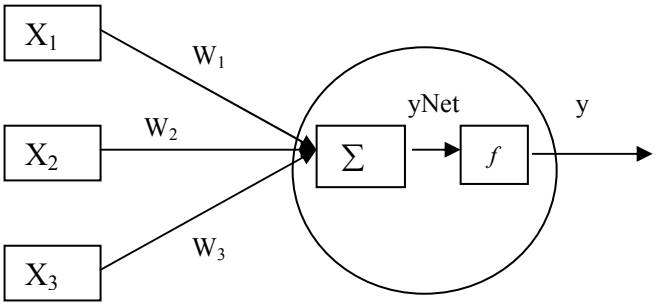
b. Cari nilai entropy setelah pemisahan dengan cara berikut :
   $$Entropy(S,A)=\sum_{i=1}^{v}(\frac{|Sv|}{S}*Entropy(Sv)) \quad (2)$$
   Dengan v adalah semua nilai yang mungkin dari atribut A, dan Sv adalah subset dari S dimana atribut A bernilai v.

c. Cari nilai *information gain* dengan cara berikut :
   Gain (S,A) = Entropy(S) – Entropy(S,A) (3)

Besarnya nilai *information gain* menunjukkan seberapa besar pengaruh suatu atribut terhadap pengklasifikasian data.

2.5 Jaringan Syaraf Tiruan

Cara kerja JST adalah meniru cara kerja otak manusia yang terdiri dari neuron-neuron. Struktur neuron pada JST digambarkan sebagai berikut :



Gambar 1 Struktur Neuron

Model matematis dari sebuah neuron adalah :

$$yNet = \sum_{i=1}^{n} x_i w_i + b \quad (4)$$

Hasil penjumlahan kemudian dibandingkan dengan suatu nilai ambang melalui suatu fungsi *threshold*. Nilainya kemudian dikirimkan melalui bobot-bobot ke semua neuron yang berhubugan dengannya.

$$y = f(yNet) \quad (5)$$

Secara umum ada 2 proses pada JST yaitu proses *training* dan *testing*. Pada proses *training*, JST diberikan pengetahuan yang berupa pola-pola data sebagai masukan untuk dilatih dan menghasilkan sebuah model JST. Pada tahap *testing*, JST akan mencoba mengenali pola-pola masukan yang diujikan untuk kemudian dicocokkan dengan model hasil dari proses *training*.

2.5.1 JST *BackPropagation*

JST *backpropagation* merupakan JST *supervised learning*, yaitu dalam proses pelatihannya memerlukan target. Disebut *backpropagation* karena dalam proses pelatihannya, error yang dihasilkan dipropagasikan kembali ke unit-unit dibawahnya.

Algoritma *backpropagation* adalah sebagai berikut :

a. inisialisasi bobot dengan bilangan acak
b. tentukan *epoch* dan *error* yang diinginkan
c. jika kondisi berhenti belum tercapai, maka dilakukan langkah d – h
d. Untuk tiap pola data *training*, lakukan langkah e – g
e. Fase propagasi maju :
   - jumlahkan semua sinyal yang masuk ke *hidden* unit

$$z\_net_j = v_{jo} + \sum_{i=1}^{n} x_i v_{ji} \quad (6)$$

   Dengan : z_net_j = total sinyal masukan pada hidden unit j
   x_i = nilai masukan pada unit i
   v_{ji} = bobot antara input unit i dan hidden unit j

   - hitung keluaran semua *hidden* unit j pada *hidden* layer

$$z_j = f(z\_net_j) = \frac{1}{1 + e^{-z\_net_j}} \quad (7)$$

   Dengan : z_j = keluaran pada hidden unit j
   Z_net_j = total sinyal masukan pada hidden unit j

   - jumlahkan semua sinyal yang masuk ke *output* unit k

$$y\_net_k = w_{ko} + \sum_{j=1}^{p} z_j w_{kj} \quad (8)$$

   Dengan : y_net_k = total sinyal masukan pada output unit k
   z_j = nilai masukan pada hidden unit j
   w_{kj} = bobot antara hidden unit j danoutput unit k

- hitung keluaran pada semua unit pada *output* layer

$$y_k = f(y\_net_k) = \frac{1}{a + e^{-y\_net_k}} \quad (9)$$

f. Propagasi mundur

- hitung faktor kesalahan pada *output* layer

$$\delta_k = (t_k - y_k) y_k (1 - y_k) \quad (10)$$

Dengan : δ_k = daktor kesalahan pada output unit k
y_k = keluaran pada output unit k

- hitung perubahan bobotnya

$$\Delta w_{kj} = \alpha \delta_k z_j \quad (11)$$

- hitung penjumlahan kesalahannya

$$\delta\_net_j = \sum_{k=1}^{m} \delta_k w_{kj} \quad (12)$$

- hitung faktor kesalahan pada *hidden* layer

$$\delta_j = \delta\_net_j z_j Z (1 - z_j) \quad (13)$$

- hitung perubahan bobotnya

$$\Delta v_{ji} = \alpha \delta_j x_i \quad (14)$$

g. Perubahan bobots

- ubah bobot yang menuju *output* layer

$$w_{kj}(t+1) = w_{kj}(t) + \Delta w_{kj} \quad (15)$$

- ubah bobot yang menuju *hidden* layer

$$v_{ji}(t+1) = v_{ji}(t) + \Delta v_{ji} \quad (16)$$

h. Hitung mse pada tiap *epoch*

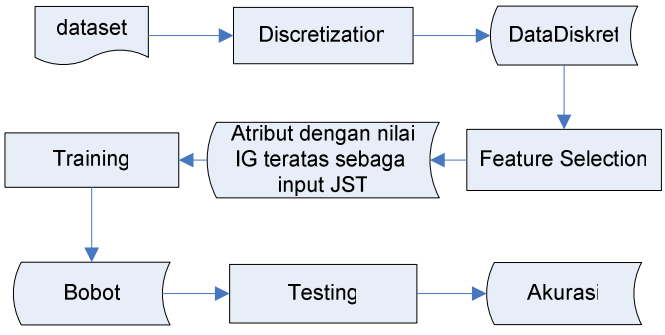
$$MSE = \frac{1}{nPola} \sum_{k}^{nPola} (t_k - y_k)^2 \quad (17)$$

Dimana : t_k = target pada output unit k
y_k = keluaran pada output unit k
n = jumlah pola

3. METODE PENELITIAN

3.1 Gambaran Sistem

Berikut merupakan gambaran sistem secara keseluruhan :



Gambar 2 Gambaran Umum Sistem

Berdasarkan gambar di atas, fungsi-fungsi yang berjalan pada sistem adalah :

a. Discretization

Fungsi ini merubah data numerik menjadi data diskret dengan menggunakan algoritma Naive Discretization. Algoritmanya adalah sebagai berikut [4] :

- pilih atribut yang akan didikretkan
- urutkan nilai pada atributnya
- identifikasi setiap data bersebelahan yang label kelasnya berbeda dan buat cut point sama dengan nilai tengah dari nilai kedua data yang bersebelahan tersebut
- hitung nilai *information gain* untuk setiap cut point yang diperoleh
- cut point terbaik untuk setiap atribut adalah cut point dengan nilai *information gain* terbesar
- lakukan langkah 2-5 untuk setiap atribut

b. Feature selection

Fungsi ini digunakan untuk menentukan nilai *information gain* dari tiap atribut, kemudian mengurutkan atribut-atributnya dari yang paling informative.

c. Konversi data

Fungsi ini digunakan untuk mengubah nilai data menjadi sesuai dengan yang dibutuhkan JST. Untuk pola input, data akan diskalakan ke dalam rentang 0 – 1 dengan menggunakan rumus :

$$x' = \frac{0{,}8(x - \min)}{\max - \min} + 0{,}1 \qquad (18)$$

d. Training

Fungsi ini digunakan untuk melakukan *traning* terhadap data latih.

e. Testing

Fungsi ini digunakan untuk melakukan uji coba sistem terhadap data uji.

3.2 Data Set

Data yang digunakan pada penelitian ini adalah data wine yang diambil dari UCI Repository yang menyediakan berbagai dataset. Data wine yang digunakan berjumlah 153 buah data terdiri dari 13 atribut. Tipe datanya adalaha numerik dan semua data memiliki label kelas. Dari keseluruhan data tersebut, data yang digunakan untuk pelatihan sebesar 78 data dan untuk pengujian sebanyak 75 data.

4. HASIL DAN PEMBAHASAN

4.1 Skenario Pengujian

Skenario pengujian yang dilakukan adalah sebagai berikut :

a. Pengujian berdasarkan perhitungan *information gain*
b. Perbandingan parameter-parameter pada JST

4.2 Hasil Pengujian dan Analisis

4.2.1 Perhitungan *Information Gain*

Berdasarkan pengujian dengan menggunakan dataset yang telah disebutkan sebelumnya, diperoleh perhitungan *information gain* untuk setiap atribut yang sudah diurutkan sebagai berikut :

Tabel 1 Hasil perhitungan *information gain* pada tiap atribut

Nama Atribut	*Information Gain*
Atr7	0,65207
Atr12	0,62848
Atr13	0,60938
Atr10	0,58005
Atr11	0,54366
Atr1	0,52692
Atr6	0,48042

Nama Atribut	*Information Gain*
Atr2	0,33905
Atr4	0,28745
Atr9	0,26240
Atr5	0,22666
Atr8	0,22341
Atr3	0,16291

Tabel di atas menunjukkan bahwa nilai *information gain* terbesar pada data Wine ada pada atribut Atr7. Hal ini menunjukkan bahwa Atr7 adalah atribut yang paling *informative*, yang artinya paling relevan terhadap kelas targetnya. Semakin besar nilai *information gain* pada suatu atribut, maka semakin besar pula pengaruhnya terhadap pengklasifikasian suatu data.

4.2.2 Hasil Pengujian Berdasarkan Jumlah Input Neuron

Berikut merupakan hasil pengujian dengan beberapa jumlah input neuron :

Tabel 2 Hasil pengujian terhadap toleransi error 0,001

Jumlah Input Neuron	Hasil Pelatihan	Hasil Pengujian	
	Epoch	Akurasi Data Latih	Akurasi Data Uji
6	199	98,7	89,3
7	242	98,7	92,0
8	130	92,3	80,0
9	129	94,8	80,0
10	142	89,7	77,3
11	144	97,4	82,6
12	174	98,7	81,3
13	160	96,1	89,3

Berdasarkan tabel di atas, menunjukkan bahwa jumlah input neuron sama dengan 7 (7 atribut), menghasilkan akurasi maksimal daripada menggunakan data dengan keseluruhan atribut. Hal ini disebabkan adanya data yang kurang *informative* pada saat penggunaan data yang lebih banyak. Pada saat penggunaan 7 atribut, yang digunakan adalah data yang paling *informative* saja. Sedangkan pada penggunaan banyak atribut/keseluruhan, data yang kurang *informative* pun digunakan dalam proses belajar, sehingga mengurangi keakuratan sistem.

4.2.3 Konstanta Momentum

Tujuan dari percobaan ini adalah menentukan konstanta momentum yang paling optimal yang dapat menghasilkan akurasi tertinggi. Hasil pengujiannya adalah sebagai berikut :

Tabel 3 Hasil pengujian berdasarkan konstanta momentum

Konstanta Momentum	Hasil Pelatihan	Hasil Pengujian	
	Epoch	Akurasi Data Latih	Akurasi Data Uji
0	276	98,71	93,3
0,05	267	98,71	92,0
0,10	242	98,71	92,0
0,15	234	98,71	90,6
0,20	223	98,71	93,3
0,25	203	98,71	94,6
0,30	237	96,15	94,6
0,35	191	96,15	92,0
0,40	223	91,02	93,3

Berdasarkan tabel diatas, dapat dilihat bahwa semakin besar nilai konstanta momentum, proses pelatihan semakin cepat. Hal ini ditunjukkan pada jumlah *epoch* yang semakin berkurang. Sesuai dengan tujuan penambahan konstanta momentum, yaitu untuk mempercepat *epoch* dengan melakukan lompatan harga mendekati posisi yang dicari. Namun, saat nilai momentum di atas 0,25 , proses *epoch* berjalan lambat, yang disebabkan karena nilainya yang besar membuat lompatan harga menjadi tidak beraturan.

4.2.4 Berdasarkan Nilai Learning Rate Awal

Untuk mendapatkan nilai awal *learning rate* yang optimal, dilakukan uji coba terhadap nilai *learning rate* awal. Hasil pengujiannya adalah sebagai berikut :

Tabel 4 Hasil Pengujian Berdasarkan Nilai *Learning Rate* Awal

Learning Rate Awal	Hasil Pelatihan	Hasil Pengujian	
	Epoch	Akurasi Data Latih	Akurasi Data Uji
0,0001	349	97,43	90,66
0,001	315	97,43	94,66
0,001	255	96,15	92,00
0,1	203	98,71	94,66
1	191	98,71	93,33

Hasil pengujian menunjukkan bahwa nilai *learning rate* awal yang paling kecil membutuhkan waktu yang paling banyak. Hal ini dapat dilihat dari jumlah *epoch* yang dibutuhkan untuk mencapai toleransi *error* adalah yang paling besar. Semakin kecil nilainya, semakin kecil pula pergeseran bobot yang dilakukan untuk mencapai toleransi errornya. Jaringan optimal saat nilai *learning rate* awal 0,1 yaitu jumlah *epoch* yang dibutuhkan relatif kecil dan kemampuan jaringan dalam mengenali data baru adalah yang paling tinggi.

5. KESIMPULAN

Berdasarkan analisis dan pengujian yang dilakukan, dapat diambil kesimpulan sebagai berikut :

a. *information gain* dapat diterapkan untuk melakukan *feature selection* terhadap sebuah data
b. 7 feature/atribut, untuk data set yang diujikan, sudah dapat mewakili data Wine dalam melakukan pengenalan
c. Konstanta momentum dan nilai learning rate menentukan kecepatan pembelajaran dalam sistem JST
d. Dengan data set yang diujikan, parameter yang paling optimal adalah dengan menggunakan konstanta momentum 0,25 dan nilai learning rate awal 0,1.
e. Walaupun tidak menggunakan seluruh informasi yang terkandung dalam data, JST *BackPropagation Momentum Adaptive Learning Rate* yang digabung dengan *feature selection*, dapat melakukan klasifikasi dengan tingkat akurasi mencapai 96%.

6. DAFTAR PUSTAKA

[1] Han, Jiawei, and Kamber, Micheline, 2000, Data Mining : Concepts and Techniques, Morgan Kaufmann Publishers, Urbana-Champaign
[2] Hermawan, Arief, 2006. Jaringan Syaraf Tiruan Teori dan Aplikasi. Yogyakarta
[3] Mitchell, TM, 1997, Machine Learning. The McGraw-Hill.
[4] Risvik, Knut Magne. DIscretization of Numerical Attributes
[5] Santosa, Budi. Data Minig Teori dan aplikasi. Graha Ilmu.